AFRL-IF-RS-TR-2005-30
Final Technical Report
February 2005

INTELLIGENT INTEROPERABLE AGENT TOOLKIT (I2AT)

Lockheed Martin Advanced Technology Laboratories

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. K748

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-30 has been reviewed and is approved for publication

APPROVED: /s/

NANCY A. KOZIARZ
Project Engineer

FOR THE DIRECTOR: /s/

JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE FEBRUARY 2005	3. REPORT TYPE AND DATES COVERED Final Dec 00 – May 04	

4. TITLE AND SUBTITLE
INTELLIGENT INTEROPERABLE AGENT TOOLKIT (I2AT)

5. FUNDING NUMBERS
C - F30602-01-C-0025
PE - 63760E
PR - AGEN
TA - T0
WU - 22

6. AUTHOR(S)
Martin Hofmann

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Lockheed Martin Advanced Technology Laboratories
3 Executive Campus, 6th Floor
Cherry Hill New Jersey 08002

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency   AFRL/IFSA
3701 North Fairfax Drive   525 Brooks Road
Arlington Virginia 22203-1714   Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2005-30

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Nancy A. Koziarz/IFSA/(315) 330-2828/ Nancy.Koziarz@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 Words)
The challenge on this effort has been how to transition current agent technology in such a way that it is widely deployed. To date, building useful agent applications for C4ISR has required significant time and money and the services of specially skilled contractors. As a result, few attempts to deploy agent applications for military use have so far occurred. The Interoperable Intelligent Agent Toolkit (I2AT) seeks to address this road block to the widespread deployment of agent technology in Command, Control, and Combat Systems by providing a means for non-technical military personnel to build an agent system, thus enabling faster decision cycles and improved decision superiority. I2AT sufficiently facilitates information agent application construction to enable domain experts to modify, construct, and interconnect applications.

14. SUBJECT TERMS
Software Agents, Agent Infrastructure, Intelligent Agents

15. NUMBER OF PAGES
21

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Table of Contents

List of Figures

Executive Summary

Under the DARPA Control of Agent-Based Systems (CoABS) program Lockheed Martin Advanced Technology Laboratories (ATL) has developed and evaluated control strategies that allow military commanders and planners to automate relevant command and control functions such as information gathering and filtering, mission planning and execution monitoring, and information system protection through the use of agent-based systems. Through the effective control of agent systems, intelligent agents work in harmony to significantly strengthen military capability by reducing planning time, automating and protecting Command and Control (C2) functions, and enhancing decision-making.

Technology advances have significantly increased the information flow to the warfighter, but increased information flow does not guarantee improved decision making, which is the true goal. Exploiting this flow requires freeing users from the myriad of information handling tasks that current C4ISR (command, control, computers, communication, intelligence, surveillance, and reconnaissance) systems impose on users so that their knowledge and skills can be used to apply information to achieve decision superiority. In several recent projects, we have demonstrated that intelligent agent technology has the potential to relieve operational personnel from the burdens of information handling tasks. Examples range from the automation of simple tasks, such as monitoring web sites for events that affect air-transport mission plans, to helping users perform difficult tasks, such as locating imagery needed for time-critical targeting (TCT).

While research continues in agent technology, others and we have amply demonstrated that the current state of the art would be of real value if placed in the hands of military users. However, agent technology is not yet in widespread use by the military. The critical challenge is how to transition current agent technology in such a way that it is widely deployed. To date, building useful agent applications for C4ISR has required significant time and money and the services of specially skilled contractors such as us. As a result, few attempts to deploy agent applications for military use have so far occurred.

The Interoperable Intelligent Agent Toolkit (I2AT) seeks to address this road block to the wide-spread deployment of agent technology in Command, Control, and Combat Systems by providing a means for non-technical military personnel to build an agent system, thus enabling faster decision cycles and improved decision superiority. I2AT sufficiently facilitates information agent application construction to enable domain experts to modify, construct, and interconnect applications. In turn, this has and will continue to facilitate transition of agent technology into day-to-day military use and prove its superior system interoperability support.

1. Background

Within command and control networks, intelligent agent technology has sparked great interest among military users, such as intelligence analysts, targeteers, and logisticians. Lockheed Martin's Advanced Technology Laboratories (ATL) and others have carried out feasibility demonstrations and experiments that have clearly shown the potential value of agent applications in military operations. This work has shown that agent technology would be beneficial across the entire spectrum of C4ISR (command, control, computers, communication, intelligence, surveillance, and reconnaissance) and combat system users.

While we have amply demonstrated that the state-of-the-art agent technology provides value if given to military users, agent technology is not yet in widespread use by the military. The challenge is how to transition current agent technology so it is widely deployed. There have been a limited number of agent-based systems that have transitioned, examples being our Domain Adaptive Information System (DAIS) to a US Army Military Intelligence Brigade and CyberXpress agent-based data mediation system, which is part of the Joint Logistics Advanced Concept Technology Demonstration, to the Defense Information Systems Agency, These and other efforts at transition have highlighted the following obstacles:

1. Applications to date have been customized to support a specific application, e.g., intelligence collection, threat identification, logistics planning, etc. Experience has shown that it is possible to build powerful applications for such problem domains but extensive development and expense are required.
2. Agent applications focused on C4ISR or combat system environments must access complex data sources and other elements of existing systems. Legacy systems are usually not designed to interoperate easily with other systems, which requires extensive analysis of interfaces and custom interface code to enable access. This drives up the cost of implementing fieldable applications because of the number and variety of legacy systems that exist within typical command and control networks.
3. Agent applications, like other C4ISR and combat systems, are likely to come from a variety of sources. Just as today's command and control networks are collections of stove-piped systems that do not interoperate well, tomorrow' networks are in danger of becoming cluttered with stove-piped agent systems.

The key to addressing these obstacles is to provide a flexible, broadly useful application package that combines:
- Core agent capabilities that users and C4ISR support personnel can easily tailor to many information-handling applications.
- Tools and application programming interfaces that enable rapid access to data sources and legacy systems.
- Agent-interoperation facilities that enable independently developed agent applications to share data, invoke each other's resources, etc.

2. Project Objective

As intelligent agent research has progressed, many agent application development and execution platforms have emerged. Some examples include Voyager, Retsina, and ATL's own EMAA. Although these platforms provide substantial development and run-time support for agent applications, it is typically necessary to do extensive development in order to create each new

application. Some research has been done in areas such as reusable agent behaviors, but current agent platforms do not provide the ability to rapidly assemble new applications from libraries of reusable components.

The overall objective of the I2AT seeks to address these issues in its effort to design, develop, and transition an Interoperable Intelligent Agent Toolkit (I2AT) that will enable users of C4ISR (command, control, computers, communication, and reconnaissance) networks and combat systems, with the aid of their information system support staff, to produce and deploy applications of intelligent, information-agent technology.

A critical part of this objective is to reduce by an order of magnitude the effort to create and maintain CoABS Grid-aware agent applications and thus to further the transition of DARPA agent technology and the CoABS Grid in particular. Beyond prototype applications, which are typically developed by the research community, successful CoABS agent and Grid technology transition must address how non-researchers may use the technology without having to be an expert in the particular field.

Our concept calls for a tool that makes creating agents as simple as defining calculations in a spreadsheet. This simplification will lead to the ability to develop complex agent-based system measurably faster and the ability to configure and modify existing agent-based applications to reflect changing operational needs and added requirements.

The I2AT leverages components and technologies from ATL's prior work in agent-based command and control applications as well as those available from the DARPA CoABS (Control of Agent-Based Systems) program, including the Agent Grid. ATL components included its database and web source adapters, query façade, database query, web retrieval, correlation, and dissemination agents.

3. The I2AT Approach
The Interoperable Intelligent Agent Toolkit (I2AT) is tailored to provide users and support personnel of C4ISR and combat system networks the ability to create and support agent information handling applications by:
• Allowing users and their support personnel to create their own application.
• Enabling support personnel to rapidly interface agent applications to legacy command, control, and combat systems.
• Enabling agent applications created by different groups to dynamically interoperate with each other, including agent applications not built with this proposed package.

This is a unique approach because it involves the user in the construction of the agent application. It removes the burden of pre-specifying domain semantics from the software developers. The I2AT is especially geared to support the creation of agent applications for discovering, disseminating, and alerting on information within C4ISR and combat system networks, as these have surfaced as the most prevalent needs in our experience with these clients. The agent applications created in the I2AT are structured as a set of agents operating in concert with a collection of servers providing data sources and computational services. This places implementation of domain specific, complex application logic in the services and allows control

3

flow of the application to be handled by the workflow of one or more agents. The I2AT supports the major steps in the agent system life cycle, including agent composition, system composition, system configuration and deployment, and system execution.

4. Technology Overview
4.1 Existing Technologies
4.1.1 LM ATL EMAA Agent Platform
The Extensible Mobile Agent Architecture (EMAA) was developed by Lockheed Martin Advanced Technology Laboratories (Figure 1). EMAA provides a rich component framework for developing or integrating distributed systems using autonomous mobile agents. The central component for EMAA is the agent Dock, which acts as a daemon process within a Java Virtual Machine (JVM) and supplies the hosting infrastructure and foundation for software agents and services. Mechanisms for reliable agent migration and authentication amongst Docks, as well as service lookup and discovery, are built into the framework.
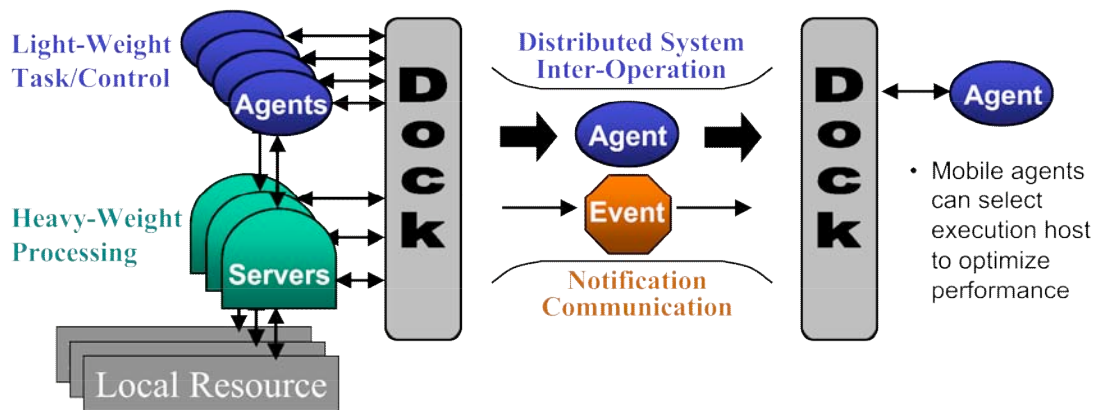

Figure 1. The EMAA Dock

A mobile intelligent agent's primary responsibility is to achieve the processing objective represented by an itinerary and assigned to it from a user or system. Mobility is an added capability, enabling the agent to relocate its processing to defined points within its itinerary. Because an agent receives an execution thread exclusively from the Dock, a mobile agent may only migrate to hosts that have an agent Dock running within a JVM. Agents often represent the business logic of an application, which may need to change based upon current circumstances. Services are components loaded into the Dock for use by agents. Typically, a service may provide a standardized interface to a resource, such as a database, or a computational engine. Reusable agent tasks represent the most common use cases of the services included. The intent of services is to encourage reusability and reduce the size of the agent when it must migrate.

EMAA has currently been incorporated as an agent architecture for more than 15 ATL contract efforts. EMAA has also been licensed for use with universities, Lockheed Martin business units, and LM ATL business partners.

4.1.2 CoABS Grid
Under CoABS, the run-time environment of our I2AT was integrated with the Agent Grid by doing the following:

- Developed tailored versions of the I2AT application development components that were targeted to C4ISR users and support personnel.
- Integrated the targeted application development components with the CoABS Grid.
- Developed additional features needed to provide a "shrink-wrapped" package.
- Established the transition process through cooperative fielding of the package to selected military users and support personnel.

This has created the kind of broad-based package that can begin the wide spread deployment of agent technology to defense users. The resulting package has enabled users and support personnel to create agent applications for C4ISR on an as-needed basis. These applications have provided automation and user assistance on information handling tasks, such as monitoring of critical events; locating, correlating, and condensing needed information; and reliably disseminating critical data. The interoperability of legacy systems has also been improved through the building of these applications.
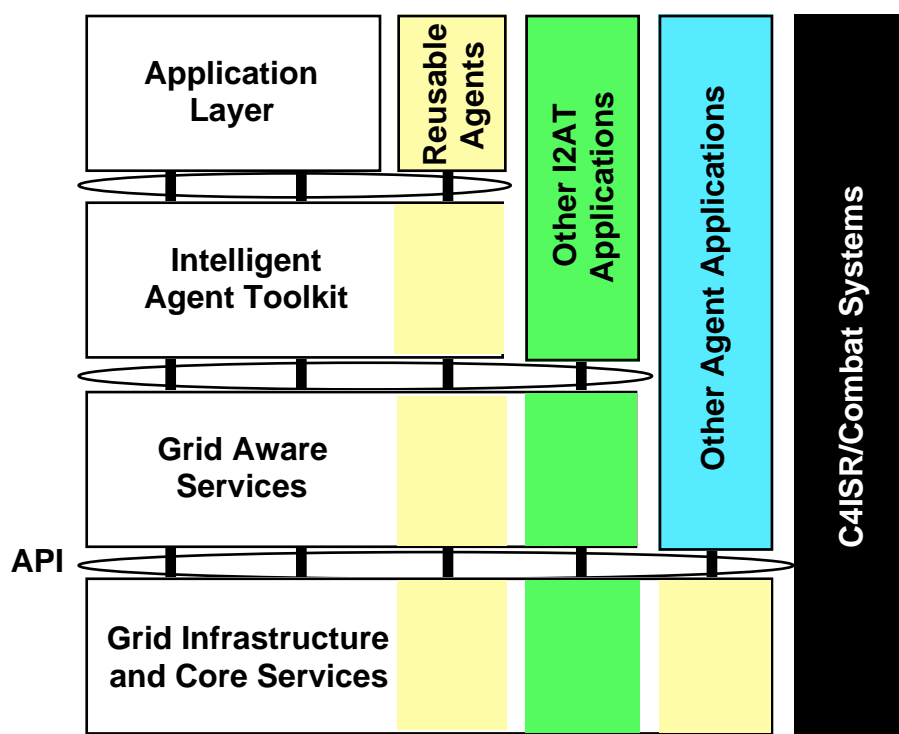

Figure 2. Interoperable Intelligent Agent Toolkit (I2AT)

4.1.3 CAST Agent-Based Decision Support Framework

CAST (Cooperating Agents for Specific Tasks) is our decision support agent framework based on EMAA and the CoABS Grid. CAST, Figure 3, provides agent behavior and cooperation patterns generally useful for decision support in C4ISR networks. CAST agents consist of Java bean-like tasks that developers compose into workflow patterns and configure for the specific installation.
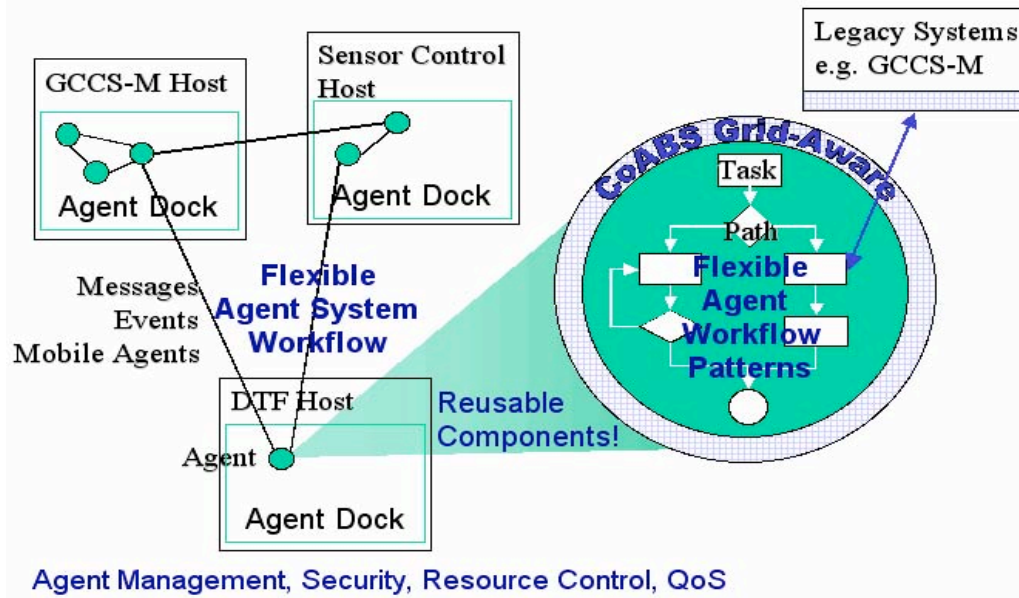
5

Figure 3. CAST Mulit-agent Framework Architecture

The CAST multi-agent framework architecture invites reuse of agent tasks, agents, and workflow patterns. It employs the CoABS Grid to ease system integration and enable interoperability.

CAST supports patterns of cooperation among specialized agents. In FBE-E we demonstrated a Theater Ballistic Missile (TBM) launch detection system composed of four diverse agent types that cooperated through a blackboard. The four agent specialties were data source monitoring, data correlation, distributed data search, and user alerting.

We are increasing the flexibility of CAST by delaying the configuration and even some composition steps until the system is installed. We have successfully used the CAST framework to tailor CAST applications for FBE-E through FBE-I, for the Joint Grid-Based Integrated Targeting (JGIT) demonstration, and for the 6th Fleet Distributed TCS Limited Objective Experiment. Figure 4 shows the Grid-supported integration of CAST into the FBE-I system architecture.

In each of these deployments, we have benefited from the interoperability provided by the CoABS Grid. We have developed Grid-based agent-service communication design patterns and stubs that normalize our solutions and accelerate our development. Using these stubs, we rapidly integrate CAST into the varying C4ISR system environments.

The CAST architecture for FBE-I highlights the benefits of Grid-supported integration. The green callout boxes show how quickly systems can be integrated via the Grid. The 3 weeks required to integrate the Modernized Integrated Database (MIDB) includes 2.5 weeks of custom development to wrap the complex MIDB data model. It took only 2 days to integrate CAST with a data mover system developed by another contractor, BTG, through which agents prompt the data mover to update specific target records. A Grid connection to a simple database reduces to a few configuration steps that we routinely complete, including testing, in a single day.
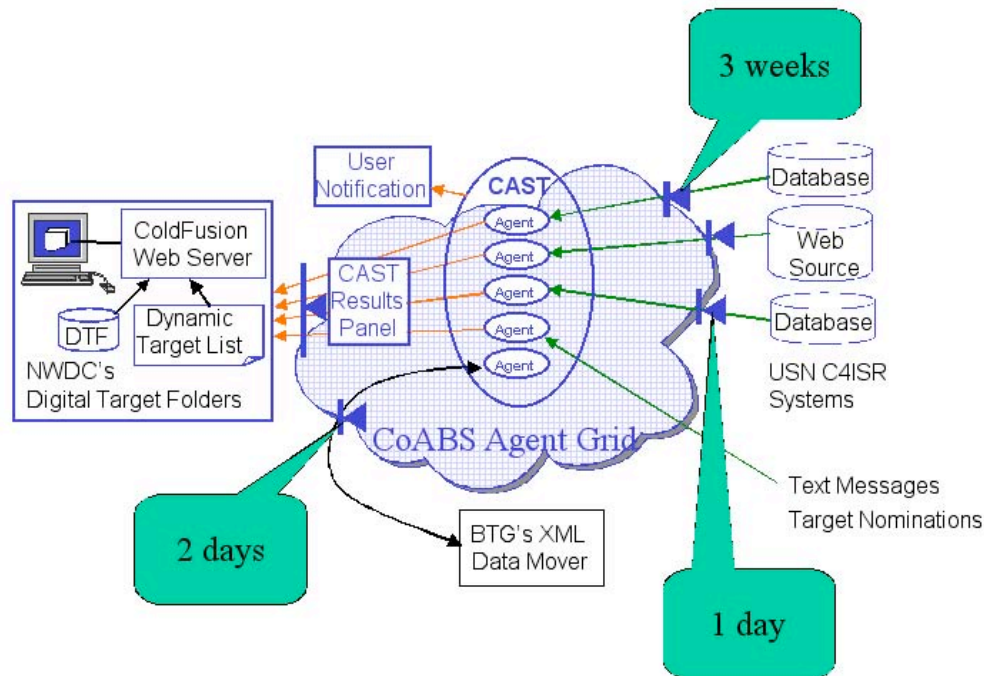
6

Figure 4. CAST Architecture for FBE-I

4.1.4 XML

Agent Work Flow Markup Language (AWFML) is an XML representation of an agent's workflow. Agent Patterns that are created in the I2AT are saved out in this format, and at run time, an agent can be launched directly from the .awfml file.

XML is also used to pass configuration parameters to an agent at time of instantiation.

4.2 Agency Oriented Development (AYOD)

From the CoABS program experience, "Agency Oriented Development (AYOD)" has emerged as a paradigm to guide the development of highly evolvable, adaptable, distributed object-oriented systems. As a new methodology, it provides rapid development and in-the-field adaptation of network-centric, distributed systems.

AYOD partitions the system space into a re*active* and *active* region. The reactive region contains functionalities that must be accessed or activated by external stimulus. These functionalities are typically services of some type.

The active region (Figure 5), contains active control logic that sequences functionalities in the reactive region to deliver system capabilities. AYOD partitions the active region into an *Agent Space* and a *User Space*.

The Agent Space contains *Software Agents* that are made up of *Tasks*, reusable elements of agent behavior. Tasks are usually lightweight controls that deliver capabilities by using one or more of the services in the reactive region.
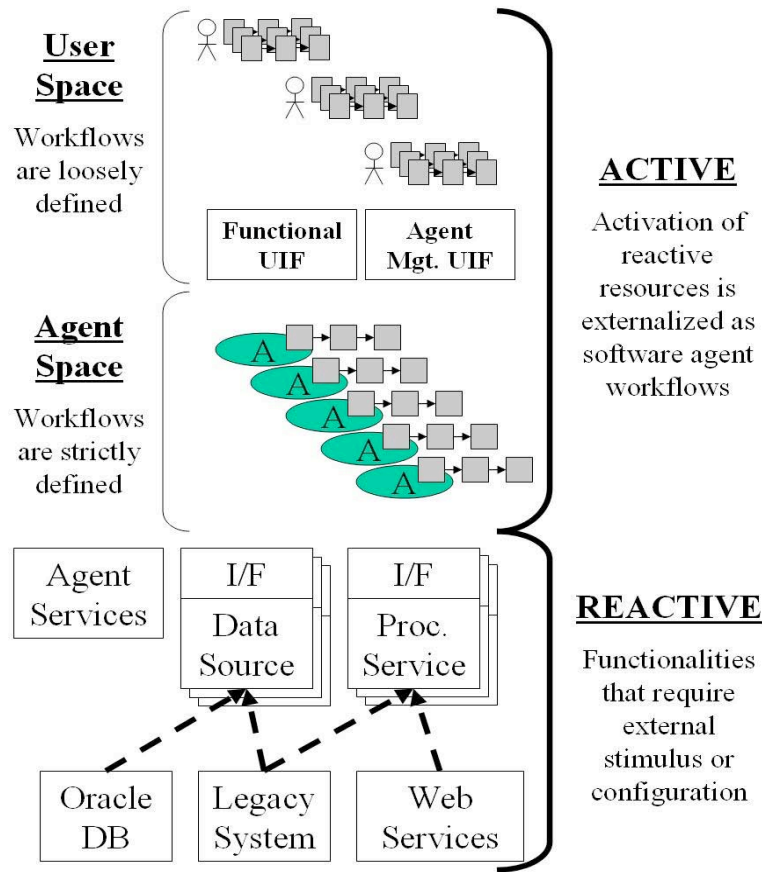
Figure 5. Active Region of AYOD

The execution structure consists of *agent workflows*, *local activities* and *execution constraints*. An agent workflow is a directed graph of local activities. A local activity is a sequence of workflow tasks. Execution constraints are policies that govern the behavior of local activities.

The User Space contains the users that use the system. Human individuals and organizations exhibit the characteristic of Agency and frequently organize functions into Workflow. Thus users are considered *Human Agents*. This conceptual similarity between the Agent Space and User Space proves useful in two ways: (1) it explicitly captures the division of function between humans and software; and (2) provides a natural metaphor for teamwork that facilitates human-system collaboration.

I2AT supports the full AYOD life cycle by providing tasks in its library that can be configured to facilitate easy access to existing services and generate at the time the system is built, the scripts and configuration files that are needed to deploy the system.

An AYOD life-cycle model (LCM) was refined and demonstrated in FBE-J for coalition C2, in the JFCOM JSM, and in FBE-I for time-critical strike. The AYOD LCM demonstrated key advantages of AYOD, including rapid development and in-the-field adaptation of network-centric, distributed systems.

4.3 Application prototype/experimentation
4.3.1 FBE-Juliet
ATL's CAST agents performed as designed and effectively supported shared situational awareness and C2 in the coalition network during the Navy Warfare Development Command (NWDC) Fleet Battle Experiment – Juliet (FBE-J) coalition experiment 24 July – 2 August 2002. CAST enabled innovative experimentation with coalition operations by synchronizing a common operational picture across the coalition C2 platforms and by enabling C2 cooperation over a robust multi-channel text message chat application.

CAST supported three FBE-J Coalition Initiative objectives:
1. It provided interoperability of heterogeneous, distributed C2 systems: the US GCCS-M, the Australian HORIZON (also used by Canada), and the U.K. MTP. CAST provided unprecedented, near real-time shared SA to all coalition partners. CAST agents allowed, for the first time, the Australian HORIZON system (also used by Canada), the US GCCS-M, and an MTP (Maritime Data Processing Tactical System) used by the U.K., to view the same tracks on their native C2 displays in near real-time. Track sharing latency was under 15 seconds under typical conditions.
2. CAST enabled dynamic reconfiguration of the system architecture. The CAST middleware allowed the vCollins to recover from communications outages that mimic those that occur while the submarine is submerged. Using CoABS Grid/Jini service discovery events backed up with a small amount of internal bookkeeping, CAST agents were notified when network connections were restored and synchronized to recover any undelivered messages. Through CAST, the vCollins received the latest track picture rapidly (about 10 seconds average) upon reconnect.
3. CAST provided a small measure of security against unintended release of information. CAST supported operator chat managed as groups over multiple channels. The CAST administrator could restrict users to groups and groups to channels, allowing for US only channels, for example. Client-side Watch Dog agents censored messages that potentially released distribution-limited information. CAST message chat helped maintain conversational context by providing complete message history to clients that joined a new channel or had been temporarily disconnected during a network outage. Such features are not available with the traditional IRC chat software.

During final installation, integration, and configuration of CAST on the coalition wide-area network, only minor final adjustments were necessary to the CAST agent software, because ATL had successfully tailored the agent system to the technical environment and the operational goals of the FBE-J coalition initiative during integration Spiral 3 in June. For example, ATL had quickly added a second method of synchronizing the COP of a system that had missed track updates due to lack of connectivity. The new method is faster because it only transfers the current picture versus the first method, which synchronizes track histories up to a chosen track age.

4.3.2 Robots, Agents, and People (RAP) Seedling
The Robots, Agents, and People (RAP) Seedling program examined the creation of new forms of highly effective, integrated, flexible teams composed of robots, agents, and people. As a new research area, the challenges ranged from understanding human teaming to determining ways

that agents and robots might adapt to a team as humans do and how they can effectively communicate with one another to solve a problem as a team. The program was designed to identify functional capabilities required for human-agent peer collaboration and to define the technical infrastructure required to provide these capabilities. The approach taken to achieve these goals was. to study military team exercises and do analysis of the requirements, lessons learned, and insights gained from these experiences.

There were three key questions raised in the study: (1) Is there a need for RAP team?. (2) Are there unique requirements for RAP teams?. (3) How much cognitive capability is required to operate in a team?. In answering these questions, the goal was to understand what drives the need for RAP teams, determine any unique requirements, formulate conceptual military teaming models, and discover their advantages and disadvantages.

Motivation for the need for RAP teams is seen in the fact that the military operates in teams. Due to different levels of expertise and aptitude, different people vary greatly in cognitive capabilities. Each performs their individual task to bring together complimentary skills to complete the overall workload. Teaming is required to achieve the depth and scope of the diversity of military tasks. Military operations require that decisions be made in a rapid, adaptive, and distributed manner. The time to execute, probability of success, and scale are greatly enhanced via teaming. A team offers diversity of ideas and multiple problem solving approaches to deal with dynamic situational issues. Teaming is required to adapt effectively and rapidly in response to a constantly changing operational environment.

The desire is for agents and robots to partner with humans, interacting as a team in the human world as a cognitive entity. Team cognition (meta-knowledge) is a key driver for individual cognition, guiding the organizational roles, policies, and procedures of the team. Agents and robots are each placed on a team based on their capabilities that fit the problem space.

Military success is increasingly dependent upon effective team performance in areas of network centric warfare, coalition operations, and operations with civilian entities such as disaster relief. It is also becoming increasing dependent upon technology to perform a variety of functions and active roles in operations, as seen in FCS, UCAR, and Objective Force Warrior.
Because the nature of the military and real world is to operate in teams, autonomous systems have proven to fall short of producing desired military operational capabilities, highlighting a need for teaming to do so. Working alone limits the mission capabilities and options. Neither is it good enough to have cognitive systems that can think on their own, they must also be able to work well with others. We place humans in roles based upon their cognitive and technical strengths and how they work with those of differing abilities. The roles that systems are capable of handling need to be evaluated in a similar manner.

Robots and agents can be viewed as tools that enhance an operators cognitive capabilities when they bring complementary cognitive capabilities to partner with humans to provide the cognitive capability to solve some larger problem that the human could not solve on their own, or we would prefer they not solve on their own given time constraints, risk factor, etc.

We cannot consider individual cognition for very long before needing to consider teaming. Teaming principals are an integral part of our thought, learning, decision, and action processes. Our understanding of an individual's contribution to a process is not evident until we see it in the context of the whole solution. Organizational learning research in the business community gives support to the statement from Jack Welsh, CEO of GE, that GE's only source of competitive advantage is the organization's ability to learn – not the individual's.

Teaming is a requirement for cognitive systems. The focus of a cognitive system is on performing a task better. It is important to learn how to perform a task within its surrounding context. That context includes others working on related tasks, who make up a team. An agent needs to be not only self-aware, but also become team-aware.

Using human-centered teaming as a foundation for RAP teams, we look at the behaviors required in military RAP teams and fit the robots and agents into human teams with no expectations that humans should need to adapt. When we view robots and agents as tactical teammates instead of human tools, they become team pears instead of something to be operated.

The use of agents was studied in six Fleet Battle Experiments, from which it was observed that agents were already acting as limited team members in C4ISR applications deployed in FBE-J. Agents and humans were seen to use the same communication means. Although agents fulfilled specific roles, pursuing their own goals, they could also be directed or requested to accommodate the needs of human team members.

We envision that the best approach to RAP teaming is a multi-disciplinary one, in which agents and agent teaming is addressed by computer science, robotics and robot teaming fall under the specialized computer science field of robotics, sociology sheds light on human teaming issues, and organizational learning is studied in management theory. By studying military models of human teaming, we see three tracks of integrated teaming analysis to be done: human teaming analysis, exploitation of research in human teaming, organizational learning, and machine learning, and the development of a computational model of teaming.

4.3.2.1 Results

From our FBE observations, it is evident that military work consists largely in teaming of related tasks. Just as not all human team members contribute equally to a job, each agent or robot may have varying levels of capabilities. Even agents with limited teaming skills can be very effective in their contribution to achieving the team objectives. It was found that certain tasks are performed best by human team members, and others by agents or robots.

Key behaviors that were identified were the exchange of information between teammates, providing support to teammates, and sharing the workload. These behaviors were modeled by the use of reasoning and interaction methods, the management of team members, and working to continuously improve performance.

The team dimension differed on each of the FBEs. Early experiments had a human team of one that was enhanced by a team of agents that retrieved information for the human teammate. Later experiments had humans teaming with agents through a dialog box, with the agents doing very

separate tasks from the humans with all control being given to the human team. The latest experiment found agents and humans collaborating on a particular task, with the agents having at least some control of their part in the process. With this structure, the humans were much more aware of the agents' contribution to the process. The main roles of the agents were to act as data collectors, data providers, and as monitors.

The utility of human-agent teams was seen in the reduction of system overload through delegation of tasks needing human direction, but not constant involvement or mere tasking. There was fast, seamless system response to known exception events from agents that monitored persistently. There was optimization of the collaboration-intensive Observe and Orient steps of the coalition C2 "OODA" loop.

Agents with rudimentary team capabilities looked like team members to users but only when the agents had well defined roles and interacted with the users. The ability to not merely task agents, but give them direction, is essential for effective teaming. It is also important for agents to be familiar with relevant parts of other team members' jobs. The value of teaming is best perceived when agents have specific roles in the team and interact directly with human teammates.

4.3.2.2 Future Directions
There are multiple questions to explore in future work in the area of RAP teaming. How much initiative should each team member take and who is responsible for delegation of tasks?. Research in the area of shared team awareness will affect the ability to adapt to changing situations, assist one another if necessary, coordinate plans and goals, dynamically reorganize the team. Deciding what, when, how, and to whom to communicate to is crucial for good team awareness. A high level task ontology is one possible way to enable a high level shared team context and the ability to reason about individual roles within a team environment, to learn and adapt.

4.3.3 CoABS Grid Extension Contributions
ATL delivered a configurable CoABS Grid-based database query component to the CoABS Grid extension release 3.3.1. The Grid extension release helped to further the transition of the Grid, a major product of DARPA's CoABS program, from research to the at large software development community. Our contribution put a powerful query capability into the hands of developers and has been useful in ATL's own system development efforts. ATL's Grid service allows users to query databases through three intuitive interfaces and performs logging, error management, and database connection management, among other functions.

ATL also packaged the agent chat application developed for FBE-J as another of ATL's contributions to the CoABS Grid extension. ATL added a ServiceUI to the client software, so that a user can download a chat client interface and participate in multi-channel chat without performing any manual software installation. Optionally, developers can install the agent-based message synchronization mechanisms from FBE-J.

4.3.4 JFCOM/J333 Joint Schedule Merge (JSM) Agent System
The I2AT life-cycle process and toolkit proved themselves on the JFCOM Joint Schedule Merger (JSM) agent tool for the JFCOM J333 Centralized Schedule Review and Deconfliction

Branch, where ATL developers built the full agent system using the toolkit's agent composer and agent system composer tools. Use of the tools enabled ATL to develop the system in a timely manner. After the demonstration at the Joint Training, Analysis, Simulation Center (JTASC), Suffolk, VA, J333 commented that ATL was the only one of their contractors who had shown a working prototype for the Joint Force Schedule Tool.

ATL developed agents that translate JTIMS, WEBSKED, and FALCON into the common format and a merger agent, which takes the translation agent's output and puts it in the final JSM flat file merged format. ATL also designed and built a Process Agent that monitors the activities of the data manipulation agents described above.

As a result of the modular agent construction provided by the I2AT, JFCOM personnel are now able to efficiently configure, maintain, and, if necessary, expand the JSM tool.

4.3.5 Expeditionary Enabling Experiments (EEE)

In 2002 NWDC decided to make agent development with the help of the toolkit one of the main EEE foci. Recognizing that ease and speed of development of agent-based systems is equal in importance to the capabilities provided by agents, NWDC designed experiments to quantify the degree of support delivered by ATL's I2AT agent development toolkit. The first experiments took place in April where ATL trained users unfamiliar with ATL's agent technology in the use of the toolkit. Then, these users made use of the toolkit to develop a number of agents, created an agent system framework, and launched the agent application. NWDC measured the time and quality of the development process. Using our I2AT toolkit, a test subject composed in an afternoon a number of functional agent systems after just four hours of instructions, substantiating the expectations of a quick learning curve and rapid acceptance of the pattern and component reuse methodology supported by the toolkit.

A second set of experiments took place in July. ATL supported the NWDC 2002 Expeditionary Pervasive Sensing (EPS) Enabling Experiments (EEE) at NUWC, Newport, RI. These experiments proved the main hypothesis guiding the I2AT toolkit effort. For two hours before the experiment, an ATL developer trained a user in using the I2AT toolkit who was experienced in Java development and the EEE sensor models, but new to agent-based systems. The test subject was given an extensive description of the software components to be used to construct the systems. This component description process consumed about an hour of time. After just 2 hours of instruction, using a tutorial ATL had developed, the test subject was given the specifications/requirements for ten systems to be created for the EEE environment. While the systems were small, they had real world applications within the EEE environment. The subject was able to construct the ten systems in approximately 4 hours. He constructed each system with little difficulty and needed only minor periodic guidance.

ATL captured a detailed account that summarizes on which tasks the test subject spent his time as he was building the various agent systems using the I2AT toolkit. ATL's analysis showed that the most time-consuming tasks are familiarization with the available agent tasks, with the software applications the agents are to interoperate with, and understanding the requirements of the system to be built. Agent composition and system composition using the toolkit are simple and quick and are rapidly learned.

4.3.6 CoAX 2002

ATL successfully integrated its contribution into the CoAX 2002 demonstration rehearsal at NWDC, Newport, RI, September 9-12. ATL achieved integration of ATL's EMAA agents with UWF's KAoS coalition policy administration and enforcement utility, with BBN's intelligent information subscriber task, and with the data consumer agents provided by U. Texas, demonstrating that ATL's I2AT toolkit framework can accommodate agent technologies other than ATL's own EMAA platform. In ATL's part of the CoAX 2002 demonstration, an ATL developer showed how the I2AT allows a user to rapidly compose and configure an agent that links a new data source into the simulated coalition command and control infrastructure. This part of the demonstration emphasized how agent technology assists in rapidly deploying and enhancing distributed coalition C2 environments.

In October the CoAX software was successfully demonstrated at NWDC in Newport, RI. The BINNI scenario of the demonstration prominently featured the Toolkit. Within the scenario, a new country joined the coalition and offered to share information from its ASW sensors to other members of the coalition. The Toolkit's ability to adapt agents and agent systems to such dynamic changes in system configuration and system environment allowed a user to rapidly construct an agent to adapt to the new data source as part of the demonstration. The resulting agent linked the new data source into the simulated coalition command and control infrastructure and information from the source was used in the remaining execution of the scenario.

5. Transition

5.1 Transition within Lockheed Martin
5.1.1 Time Critical Target Dynamic Decision Enabler (TDDE)

The primary goal of the TDDE effort was to identify and develop automated support tools for assisting the TCT Cell Chief in managing the prosecution of TCTs in under a two (2) minute time frame. This task currently requires more than a dozen Air Force personnel with the correct experience and training, and their decisions are based on available Essential Elements of Information (EEI). A significant part of ATL's contribution to this effort was in supplying agents that were built with the I2AT to extract the EEI in a timely manner and provide notification to the TCT Cell Chief.

5.1.2 Intelligent Data Analysis Capability (IDAC)

Web Accessible Agent Templates (WAAT) uses agent workflows that have been built using the I2AT to generate agent templates that publish selected attributes to be made available for modification by an end-user. These templates are used to generate a web interface that allows non-developers to configure the agent template, which can then be launched as a custom agent from the web page. The interface also allows the user to monitor the progress of running agents that have been launched. The underlying structure of. the I2AT is used in launching and monitoring the agents.

5.2 Transition to Customers
5.2.1 LOE within the EEE

ATL coordinated with NWDC to conduct an LOE within the EEE initiative. Funded by ONR, the LOE examined advanced network communications, collaborative tools, and agent based

computing in support of a distributed command and control staff. The experiment was conducted during the COMSECONDFLT Multinational Maritime Exercise (MNME) in Norfolk, VA, from 7-11 April, 2003. A primary objective of the MNME was to examine new processes for coordinating command and control when a Joint Force Maritime Component Commander's (JFMCC) staff is disparately located, e.g. split between a forward/afloat and rear/ashore position.

For the experiment, LM ATL developed and deployed an agent-based application aimed at improving split staff situational awareness during the daily battle rhythm execution process. The agent-based application allowed the battle watch captain and other JFMCC components to assess battle rhythm progress, address obstacles to a smooth battle rhythm, and rapidly notice and react to unexpected events affecting the rhythm. The application, called the Dynamic Staff Situational Awareness Tool, provided the abilities to assess the state of overall rhythm progress, to assess the progress on selected rhythm events, to view and modify progress/details of input and output products for a selected battle rhythm event, and to view a list of the participants in online conferences related to specific battle rhythm events. The application also provided spoken and visual alerting capabilities for notification of items of interest. A graphical representation of the Dynamic Staff SA Tool as shown in Figure 6.
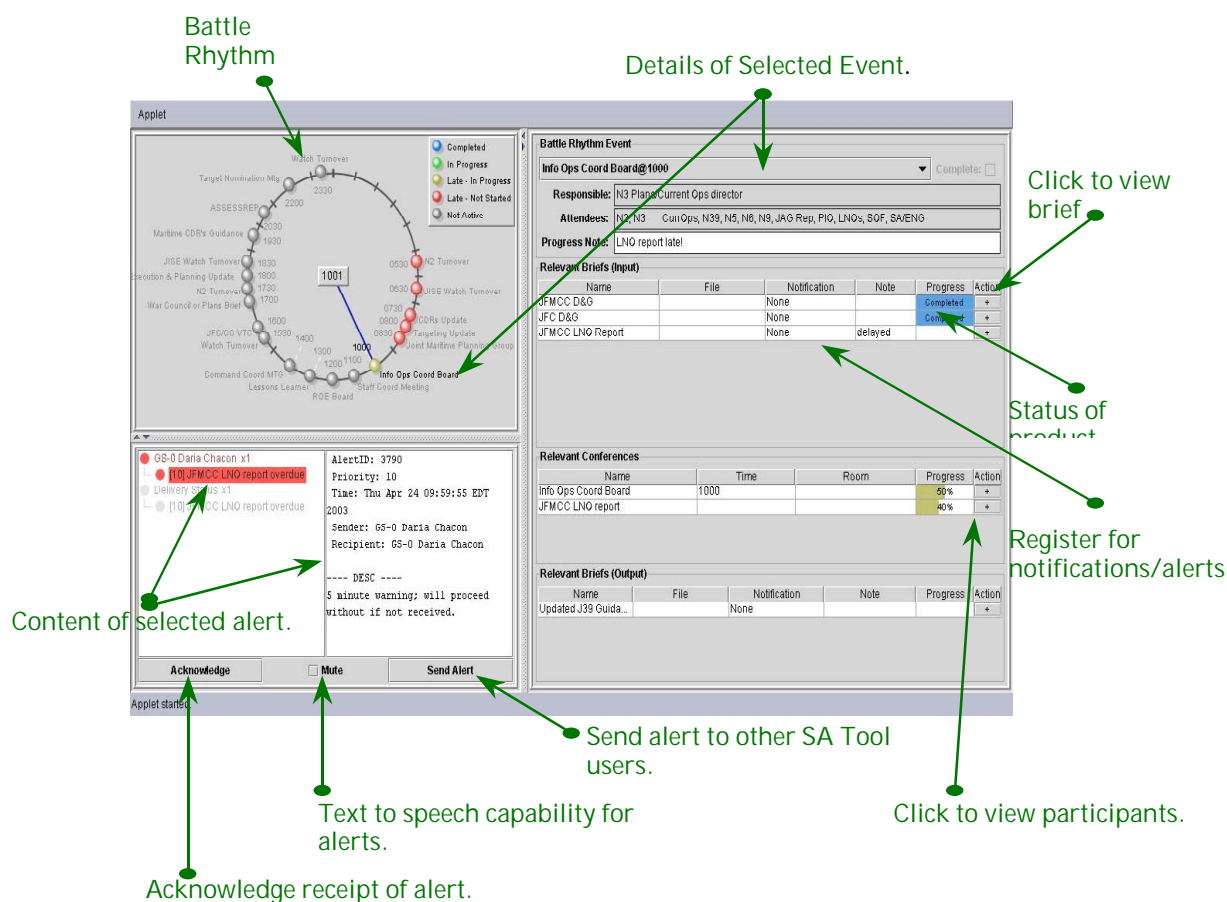

Figure 6. The Dynamic Staff SA Tool.

15

The Dynamic Staff SA tool received significant attention and acclaim both at the operational and command levels for its potential to improve situational awareness into the battle rhythm execution process and reduce the amount of time that seasoned officers spend performing tedious tasks in support of the battle rhythm. During the experiment, the Battle Watch Captain asked that the Dynamic Staff SA Tool be displayed on a large plasma monitor at the front of the Joint Operations Center. The Tool also had excellent exposure to and feedback from Senior Commanders (VADM Dawson of C2F, VADM Mayo of NETWARCOM, RADM Route of NWDC, RADM Clark of C2F, CAPT Squicciarini of TACTRAGRULANT, etc.). The C2F/NWDC QuickLook report for the LOE offered the following comments on the Dynamic Staff SA Tool.

> *"The operations staff and Battle Watch Captain (BWC) watchstanders found the battle rhythm SA tool useful in the ability to monitor the progress of the battle rhythm events in near-real time, while reducing the non-value added activity of finding, tracking, retrieving and displaying briefs and status reports. The SA tool generated interest at high levels and offers a potential solution for the complexity of information sharing in a distributed environment."*

The Dynamic Staff Situational Awareness Tool was also an integral component of the Assessment of the Effectiveness of Intelligent Agent Decision Support effort also conducted by LM ATL during the MNME. The Assessment of the Effectiveness of Intelligent Agent Decision Support, funded by Office of Naval Research (ONR), tested the hypothesis that intelligent agents improve operational decision making speed and quality. The assessment effort consisted of three primary activities: development of the assessment experiment plan, collection of data during the execution of the experiment, and analysis of the collected experiment data. The Dynamic Staff Situational Awareness Tool served as the agent-based application with which the hypothesis was tested.

In early 2004, ATL participated in a Limited Objective Experiment (LOE) with the Second Fleet to test how agent-based software could assist the J2 (Intelligence) with their daily intelligence collection activities. As part of the experiment, ATL developed a Chat Task Assistant using I2AT technology that allowed an intelligence officer to monitor chat channels for items of interest. In April that software was deployed on the USS Mount Whitney in anticipation of utilization during a Joint Task Force Exercise (JTFEX) in June.

6. Dissemination of Results
ATL has publish papers describing how our I2AT technology has been used. The following make up that list.

"Agents and Robotic Teaming With People – A New Strategy for Cognitive Systems Research," Whitebread, K., Pridmore, L., Hofmann, M., Buczak, A., 7th World Multiconference on Systemic, Cybernetics, and Informatics, Orlando, FL, July 27-30, 2003.
"Agency-Oriented Development: A Methodology for Developing, Maintaining, and Adapting, Systems of Systems," Lardieri, P., Whitebread, K., Orr, M., in process of being published, 2004.

"Performance Analysis of Mobile Agents for Filtering Data Streams on Wireless Networks,"
Kotz, D., Cybenko, G., Gray, R., Jiang, G., Peterson, R., Hofmann, M., Chacon, D.,
Whitebread, K., Hendler, J., to appear in Mobile Networks and Applications, Vol. 7, 2002.

"Mobile-Agent versus Client/Server Performance: Scalability in an Information-Retrieval Task,"
Gray, R., Kotz, D., Peterson, R., Baraton, J., Chacón, D., Gerken, P., Hofmann, M.,
Bradshaw, J., Breedy, M., Jeffers, R., Suri, N., in Proceeding of the 2001 National Fire
Control symposium, Lihue, HI, August 12-30, 2001.

"CAST Agents: Network-Centric Fires Unleashed," Martin O. Hofmann, Daria Chacón, Gerard
Mayer, Kenneth R. Whitebread, James Hendler, in Proceedings of the 2001 National Fire
Control Symposium, Lihue, HI, August 12-30, 2001.

"Intelligent Mobile Agents in Military Command and Control," McGrath, S., Chacón, D.,
Whitebread, K., Fourth International Conference on Autonomous Agents, Workshop on
Agents in Industry, Barcelona, Spain, June 3-7, 2000.

"Mobile Agents on the Digital Battlefield," Hofmann, M., McGovern, A., Whitebread, K.,
Second International Conference on Autonomous Agents, Minneapolis, MN, May 10-13,
1998.

7. I2AT Follow-on

Recent developments in semantic mark-up allow us to describe and advertise services in
registries that are accessed by potential users. ATL seized the opportunity to reconsider the I2AT
agent representation and agent launching capabilities, in order to leverage the expanding set of
tools and standards of the Semantic Web. ATL implemented an initial infrastructure for a
Semantic Web-based version of I2AT based on the existing, proprietary I2AT agent description
and toolkit code. We developed the AgeNt Itinerary Markup Language (ANIML), a language
conforming to OWL-S, to describe agent workflows. ANIML describes an agent template as a
composite Web service. By making these ANIML templates available in a registry service, a user
can find an agent that performs a general task and configure it to perform the work exactly as
desired.

ANIML has several advantages over our previous XML representation. ANIML is not
implementation specific. It simplifies agent configuration, since it has the capability to limit or
expand the scope of task parameters in an agent workflow. OWL-S is also used to describe each
task that is used in the workflow like a Web service. We developed a proof of concept
demonstration that instantiates and runs a DatabaseQueryAgent that queries a user specified
database, persisting until results are found, and then displays the results.

The ANIML file for our demonstration was hand built, but we envision a future Web-based
version of the I2AT that will build the ANILML file by accessing task descriptors (OWL-S
descriptions of tasks that can be made available to the toolkit) to provide the building blocks and
user interface for one to create an agent workflow and save it out as an agent template in the
form of ANIML. The toolkit would also allow a user to find pre-built agent templates and
configure them.

We plan to carry this technology forward by developing Agent Wizards based on this
modernized infrastructure for the DARPA FastC2AP program and potentially other venues, such
as LCS.